

Design, Validation and Prototyping of the EMS SDH STM-1 Mapper Soft-core

Ney L. V. Calazans¹, Fernando G. Moraes¹, César A. M. Marcon², José C. S. Palma²
{calazans, Moraes}@inf.pucrs.br, {marcon, jcsalma}@inf.ufrgs.br

1 Pontifícia Universidade Católica do Rio Grande do Sul – PUCRS/FACIN
Av. Ipiranga, 6681 - Prédio 30 / Bloco 4 - 90619-900 - Porto Alegre - RS – BRAZIL

2 Universidade Federal do Rio Grande do Sul – UFRGS/PPGC
Av. Bento Gonçalves, 9500 - Prédio 43412 / Bloco IV – CEP: 91501-970 - Porto Alegre - RS – BRAZIL

Abstract

This paper describes the design, verification and prototyping of EMS, a telecommunication intellectual property soft-core developed in the scope of an industry-academia cooperation. EMS performs insertion (mapping) and extraction (demapping) of E1 channels into/from Synchronous Digital Hierarchy (SDH) frames. The paper addresses the solution of several synchronization problems implied by the E1 channels mapping/demapping process. EMS was fully described in RTL VHDL. It was functionally validated by simulation and prototyped in FPGA platforms. Together with the exploration of the techniques involved in embedding E1 into SDH frames, another contribution of the work is the availability of a reusable and parameterizable telecom core with high performance, low latency, and small size.

Keywords: SDH-E1 mapping/demapping, soft IP core.

1. Introduction

Over the last three decades, communication networks are evolving from analog to digital [1], which results in better transmission quality and larger bandwidth. In many areas, this technological change gave a boost to telecommunication systems research. With the advent of the Web comes an unprecedented increase in data traffic. The telecom systems complexity is growing faster, due to: (i) media and protocol diversity; (ii) applications nature variety; (iii) increased data communication speed; and (iv) high volume of data to be transmitted. To cope with these fast changing scenarios, new telecommunication technologies together with design, validation and test techniques are necessary. The Synchronous Digital Hierarchy is one example technology, due to its high data communication speed and high capacity to pack telecom protocols with different natures, like E1, ATM and others.

This work introduces EMS, an E1 channel Mapper/Demapper into/from SDH frame. EMS is an intellectual property (IP) soft-core developed in the scope of industry-academia research and development cooperation. EMS is a successor of an E1 IP soft-core design [2]

developed in the same cooperation¹. An IP core is a pre-designed and pre-verified module used in combination to compose larger circuits, typically custom VLSI integrated circuits or large programmable devices, such as multimillion-gate FPGAs.

E1 carriers have a definition originated from the Plesiochronous Digital Hierarchy (PDH). PDH are communication systems where transmitted signals have the same nominal digital rate but are synchronized with different clocks. PDH is a hierarchy employed in data and voice transmission systems with plesiochronous synchronization. PDH is also a conventional multiplexing technology for network transmission systems. E1 is a PDH base standard used in Europe and South-America, with a transfer rate of 2.048 Mbps. North America and Japan employ the T1 or DS-1 base standard, with a transfer rate of 1.544 Mbps.

The goal of this work is to present EMS design and verification. Its main contributions are: (i) the design of a soft-core compliant with a specific ITU-T standard [3]; (ii) the prototyping of the system in hardware to guarantee timing constraints using Virtex FPGAs; (iii) the verification process of EMS, which requires a huge volume of data to validate its functionality; (iv) the accurate buffering analysis to perform E1 mapping and demapping into/from SDH frames; (v) the availability of a design with small footprint, which allows straightforward scaling and IP reuse in larger circuits.

This work is organized as follows. Section 2 explains some basic concepts of SDH. Section 3 presents the state-of-the-art in designing SDH mappers. Section 4 introduces the EMS architecture. Section 5 describes the EMS validation. Section 6 describes the EMS prototyping. Section 7 presents some conclusions.

2. Synchronous Digital Hierarchy

SDH employs synchronous time division multiplexing techniques to transmit different tributaries (E1, Ethernet, ATM, etc) through the same physical channel. A primary goal in the development of the SDH formats

¹ This work was developed with support of Parks Comunicações Digitais in the FACIN/PUCRS.

is to define a synchronous optical hierarchy with sufficient flexibility to carry payloads of different types. SDH basic modular signal is called Synchronous Transport Module level one (STM-1), which operates at 155.52 Mbps. Synchronous networking differs from PDH in the exactness of data transport rate. SDH systems are tightly synchronized to network base clocks, making the entire network operating synchronously.

The SDH is a multiplexed structure. Different containers (C-12 and others) with different rates are mapped to virtual containers (VC-11, VC-12, VC-2, VC-3 and VC-4). Pointers implement virtual container alignment, generating tributary units (TU-11, TU-12, TU-2 and TU-3) or administrative units (AU-3 and AU-4). Tributary units are multiplexed in tributary unit groups (TUG-2 and TUG-3) according to container rate. TUG-2 can be multiplexed in VC-3 or TUG-3, and TUG-3 is multiplexed in VC-4. Administrative units are grouped in administrative unit group (AUG). Finally, AUG is multiplexed in one or more STM-1s.

Figure 1 exemplifies one possible composition of the STM-1 frame structure. This composition reflects how EMS operates. Each STM-1 frame has 9 rows and 270 columns. Each column has a width of 1 byte.

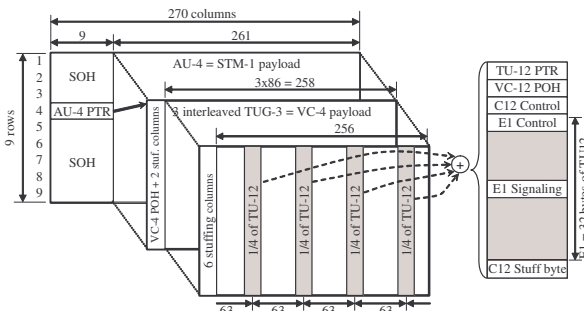


Figure 1 – SDH STM-1 frame structure.

The first nine columns are transport overhead, combining a pointer (AU-4 PTR) and section overhead (SOH). SOH contains framing error monitoring and management. AU-4 PTR identifies the VC-4 start point. STM-1 payload or AU-4 uses the remaining 261 columns. The first three VC-4 columns are VC-4 path overhead (POH) and two stuffing columns. Three interleaved TUG-3 are mapped in the remaining 258 columns of VC-4 (VC-4 payload). The VC-4 payload is composed by 6 stuffing columns and 63 interleaved TU-12s. Each TU-12 is distributed along four columns, summing up a total of 36 bytes (9 bytes per column). The VC-12 virtual container preceded by a POH forms a TU-12. As a consequence, each VC-12 is composed by an E1 carrier plus two stuffing/control bytes.

Each frame transports 19,440 bits (270 x 9 bytes) at rate of 155.52 Mbps, implying a frame period of 125 μ s. Four frames compose a super-frame (SF), which is the highest structural level of SDH STM-1. As depicted in Figure 2, the STM-1 payload pointed by the J0 byte may float inside the STM-1 frame. J1 marks the start point of each VC-4 payload inside the STM-1 payload.

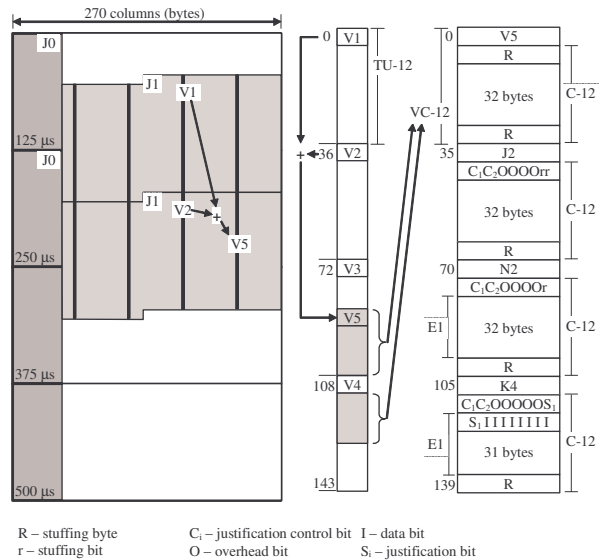


Figure 2 – SDH-STM1 SF structure with VC-12 mapping.

In each SF, there are four different kinds of TU-12 PTR (V1, V2, V3, and V4). V1 and V2 pointers are joined to compose the V5 address, which marks the beginning of C-12. When PDH and SDH clock frequencies are synchronized, there are 1024 bits of useful data transmitted in 4 E1 channels. Otherwise, it is possible to add or subtract one bit from each SF. The C-12 structure of SF merges two/three extra bytes with 32/31 E1 bytes. The first C-12 encloses two stuffing bytes and an E1 channel. The second and the third C-12 of each SF contain one control byte, one stuffing byte and one E1 channel. The last C-12 of each SF encloses one control byte, seven data bits plus a justification opportunity bit, one stuffing byte and 31 bytes of E1 channel. The majority of C₁s and C₂s bits imply positive or negative frequency justifying. It means that when two or more C_i bits are 1, the S_i bits are valid data. This, in turn, means that 1025 bits of data are available for SF.

On the other hand, if two or more C_i bits are 0, the S_i bits are stuffing bits, meaning that 1023 bits of data are available for SF. The number of bytes implies slight changes of clock frequency. For instance, 1025 bits implies 2,050 KHz and 1023 bits implies 2,046 KHz. This frequency change allows mapping/demapping a PDH frame into/from an SDH frame without data loss.

3. Related Work

The demand for telecommunication services leads to a significant offer of SDH systems in the market [4][5]. Each distinct equipment presents its own distinguishing features, different prices, and use cases.

For instance, Yongming et al. [6] consider three ways to mapping asynchronous 2.048 Mbps tributary into SDH VC-12: asynchronous, bit synchronous and byte synchronous. They focus on the asynchronous mapping, discussing positive/zero/negative justification to improve the capacity of elastic buffer store. Fuqiang et al. [7] make similar analyzes, quantifying the best

elastic buffers size for PDH to SDH and SDH to PDH conversions.

Clauberg et al. [8] introduce a scalable modular architecture for SDH technology, by exploiting the regular multiplexing principle inherent to the SDH hierarchy. They demonstrate the feasibility of their scalable modular architecture with a framer chip, able to handle 4 x STM-1 and variations of STM-4. Peng et al. [9] developed THXC, an SDH cross-connected ASIC with an embedded built-in self test circuit. THXC is programmable and monitored by an external computer. It was designed to allow many kinds of switching rates and flexible cascadability.

Silveira and van Noije [10] present the modeling of an E1 mapper for SDH Systems, pointing out the difficulties to implement these systems due to the synchronization mechanisms and the PDH information nature that are carried in SDH frame.

This work introduces EMS architecture, which is similar to Peng et al. [9] system with the scalability introduced by Clauberg et al. [8]. In addition, the approach is distinct from that of Fuqiang et al. [7], in order to achieve better elastic buffers sizing for PDH to SDH mapping and vice-versa. It also shows the functional verification process to deal with the huge amount of input/output data.

4. EMS Architecture

EMS is a scalable architecture, allowing from 1 to 63 E1 channels mapping/demapping into/from an SDH frame. The smallest functional EMS operates with one E1 channel and is called *Basic EMS*, or simply BEMS.

The BEMS external interface is implemented by three signals sets. The first set is composed by `RST_N`, `CK32_768` and `CK65_536` signals, used to perform the system control and synchronization.

The second set is composed by `DTBYCK`, `DTBPAY`, `DTBJOJ1`, `DTBDATA` and `DTBDATAOUT` signals. These signals partially implement a Telecom Bus Interface. Telecom Bus is a byte-wide parallel format that serializes and deserializes data and identifies the J0 and J1 pointers in the SDH frame, reducing the core operating frequency from 155.52 MHz to 19.44 MHz [4].

The third set is composed by `E1IN`, `CKE1IN`, `E1OUT`, `CKE1OUT` and `CHANNEL` signals. These signals implement the E1 interface. To implement E1 channel mapping into SDH frames, the EMS core adds data or voice from an E1 channel (`E1IN` signal) to the Telecom Bus (`DTBDATAOUT` signal). The E1 demapping from SDH is implemented by receiving data or voice from the Telecom Bus (`DTBDATA` signal), mapping these to an E1 channel (`E1OUT` signal). The selected E1 channel is addressed by `CHANNEL` signal, which allows specifying one out of 63 channels.

Five main modules and an auxiliary logic circuit compose BEMS system, as Figure 3 shows: *ColumnAddress*, *Delay*, *V5Enable*, *VC12Drop* and *VC12Add*. The last four modules are encapsulated in a module called

AddDrop. This hierarchical structure confers a high scalability to EMS. To increase the number of E1 channels to n it is just necessary to instantiate the *AddDrop* module n times plus an extra structure to control `DTBDATAOUT` multiplexing.

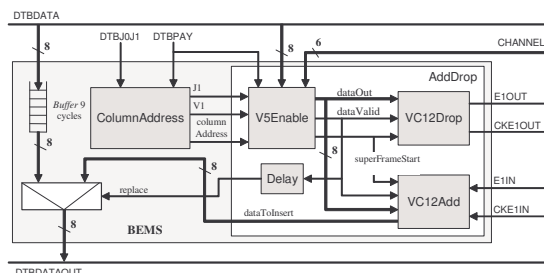


Figure 3 – BEMS internal modules organization.

4.1 ColumnAddress Module

The *ColumnAddress* module accepts as inputs the Telecom Bus control signals and generates the J1 pointer (payload start), V1 pointer (TU-12 start in the Telecom Bus) and the number of each column present in the payload (through the `columnAddress` signal). The column number allows the system to find the data of the required channel in the Telecom Bus. The J1 pointer corresponds to the first column number. The V1 pointer allows the system to locate the VC-12 internal pointers (e.g. V5 pointer).

4.2 Delay Module

The *Delay* module generates the `replace` control signal, responsible for defining the correct moment to insert data in a valid VC-4 column.

4.3 V5Enable Module

The *V5Enable* module searches for data valid (`dataValid` signal) in the Telecom Bus. It also indicates the super-frame start (`superFrameStart` signal). In addition, the *V5Enable* module stores data from Telecom Bus and forwards them to *VC12Drop* and *VC12Add* modules. The module also detects valid columns through a table mapped into a ROM-like structure. Based on the first occurrence of each channel, the *V5Enable* module computes the others three TU-12 occurrences in the VC-4 payload by adding 63, 126 and 189 to the first column value. To extract VC-12 from TU-12 it is necessary to remove the bytes corresponding to V1, V2, V3 and V4 pointers that are represented by TU-12 PTR in Figure 1. These 4-byte positions are also provided by the `columnAddress` signal. The data is valid only if its address is different from the pointer addresses (1, 36, 72 and 108, respectively). In this case, a valid data is provided through the `dataOut` signal.

The V5 pointer address (represented by TU-12 POH in Figure 1) is computed joining the two last significant bits of the V1 pointer and the eight bits of V2 pointer. This results in an offset, as shown in Figure 2. The VC-12 address is counted from V2 pointer. For example, if the address of V5 pointer is 0, it means that this pointer

is located in the first byte after the V2 pointer. The pointers V1, V2, V3 and V4 must be skipped. When the offset is between 0 and 34, it means that V5 starts after V2 and before V3 pointers, being necessary to add 37 to the offset, to compose the V5 pointer address. When the offset is between 35 and 69, it is necessary to add 38 to the offset, skipping the bytes above the V3 pointer. The last case is when the offset is between 105 and 139. In this case, the offset must be decreased by 104. The resulting address will be between V1 and V2 pointers.

4.4 VC12Drop Module

The *VC12Drop* module is responsible for extracting data from the Telecom Bus and sending these to the E1 output channel. This module has an internal 64-bit circular FIFO buffer. The write pointer starts pointing to position 0. To avoid data loss, the FIFO is read only when half of it is written. In other words, when the write pointer reaches the position 32, the system starts the drop operation. After this, the FIFO data reading is continuously active, and the reading clock is adjusted according to the difference between the write and read pointers expressed by the DELTA signal. This operation performs zero/negative/positive frequency justification. If DELTA is greater than the FIFO length, there is some data loss. If DELTA is too small, the system can drop wrong data. The FIFO is dimensioned to avoid these problems.

A hysteresis mechanism was implemented to control the variation of the DELTA signal. It allows keeping a minimal and a maximal distance between read and write pointers, before executing a positive or negative justification. Figure 4 shows this behavior. When the DELTA signal is between the minimal and maximal value, the system operates at the nominal frequency (2.048 MHz). To obtain a 2.048 MHz clock, the 65.536 MHz reference clock is divided by 32. When the DELTA signal reaches the maximal hysteresis value (48), the reading frequency (CKE1OUT signal) must be increased, and when the DELTA signal reaches the minimal hysteresis value (16), the reading frequency must be decreased.

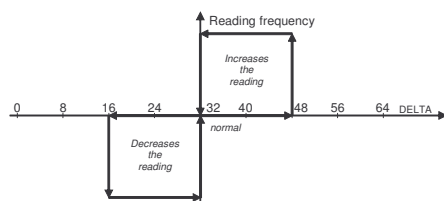


Figure 4 – Hysteresis behavior for VC12Drop.

As depicted in Figure 2, the control justification bits (C_{1S} and C_{2S}) indicate if the bits S_1 and/or S_2 are valid data bits. When the justification bits indicate valid data, the bits contained in S_1 and/or S_2 must be written in the FIFO. In the nominal frequency, only one of them is valid data. When S_1 and S_2 are valid data bits, the amount of data to write in the FIFO is increased, and the DELTA signal is also increased, reaching the maximal hysteresis value. In this case, the reference clock is divided by 31, to obtain 2.114 MHz frequency. This

higher frequency reduces the DELTA signal to the normal range and consequently the nominal frequency is recovered. When S_1 and S_2 are not valid data bits, they are not written to the FIFO, making the DELTA signal reach the minimal hysteresis value. In this case, the CKE1OUT frequency must be decreased, dividing the reference clock by 33, to obtain a 1.986 MHz frequency. This lower frequency increases the DELTA signal to the normal range and as a result, the nominal frequency is recovered.

The FIFO and hysteresis limits have to be dimensioned to avoid data loss, increased latency and memory usage. A very short FIFO can cause data loss, since bytes are written in burst using the 19.44 MHz clock frequency, while reading is continuously performed at a 2.048 MHz clock frequency.

A very large FIFO can increase memory cost and latency, and lead to improper operation. This is due to the time that the output frequency stays different from the nominal. Even with good FIFO dimensions, the circuit may operate improperly due to the difference between minimum and maximum hysteresis limits. If the limits are too near or too far to/from each other, the output frequency will change too fast or too slow violating the ITU-T standard. This may, in turn, damage the signal recovery by an external E1 processing module.

Figure 5 shows the DELTA signal behavior for nominal clock operation, considering a FIFO with 64 bits, and maximum, medium and minimum hysteresis limits of 48, 32 and 16, respectively. The analyses confirmed that the chosen hysteresis limits are the best values to respects all ITU-T rules.

4.5 VC12Add Module

The function of the *VC12Add* module is to insert data from an external E1 channel in the Telecom Bus. These data are received at CKE1IN rate, an operating frequency that may vary. According to this variation, the *VC12Add* module executes the frequency justification through S_1 and S_2 justification opportunity bits, and C_1 and C_2 justification control bits. The justification frequency is based on a hysteresis mechanism, analogous to the one used by the *VC12Drop* module. For data insertion, a 128-bit FIFO is needed, with minimum and maximum sizes of 32 and 96, respectively. The FIFO size of the *VC12Add* module is bigger than the corresponding FIFO of the *VC12Drop* module. This occurs due to the results of the analysis of worst case synchronization conditions between PDH to SDH. These results pointed that during the Add operation the number of bits inserted may vary more widely than during the Drop operation.

When the PDH and SDH clocks operate at their nominal frequencies, only one of S_1 or S_2 is used as valid data bit. When the CKE1IN frequency is higher than the nominal value, the amount of data written into the FIFO is increased and the DELTA signal reaches the maximum hysteresis value. This increases the reading rate of E1 data input and adds to the amount of data into

Telecom Bus, avoiding data loss. These extra Telecom Bus data must be inserted into S_1 and S_2 bits and the justification control bits must be set to 1. When the $CKE1IN$ frequency is lower than nominal, the amount of data written into the FIFO is less than the amount of data read. As a result, the $DELTA$ signal reaches the minimum hysteresis value. To avoid incorrect readings, it is necessary to decrease the Telecom Bus reading speed. Thus, S_1 and S_2 bits are not filled with valid data and justification control bits are set to 0.

Figure 6 shows $DELTA$ signal behavior for $VC12Add$, considering nominal (2.048 MHz), high (2.050 MHz) and low (2.046 MHz) E1 input clocks. With $CKE1IN$ low frequency, there are more data bits processing than E1 input data bits generation. To avoid incorrect readings, $VC12Add$ reduces the number of valid data bits in each super-frame to 1023. The opposite happens when the E1 input clock ($CKE1IN$) has a frequency higher than nominal. To avoid data loss, $VC12Add$ increases the number of valid data bits in each super-frame to 1025.

5. Functional Validation

One major difficulty with the EMS functional validation step is the number of simulation cycles needed to verify each design aspect, and the huge amount of data produced and consumed during a simulation, even to provide a moderate covering of the design features. For instance, to evaluate 1 ms of real operation it is necessary more than 155 thousand bits of data. To minimize the problem, external software was written. A parameterizable *pattern generator* creates verification multi-frames according to the parameters and the circuit under verification. A *pattern analyzer* compares simulation results (output files) against the input files and input parameters. This software was implemented for both PDH and SDH sub-circuits.

The validation process was conducted in three scenarios of increasing complexity:

- i. The first scenario is a *loop verification* that evaluates the Telecom Bus (SDH input/output) and *AddDrop* (PDH input/output) circuits separately. The Telecom Bus circuit corresponds to the left side of Figure 3, comprising the buffer, *ColumnAddress* module and multiplexers, while *AddDrop* circuit corresponds to the right side of Figure 3. It comprises modules *VC12Drop*, *VC12Add*, *Delay* and *V5Enable*;
- ii. The second scenario, *BEMS verification*, evaluates the mapping/demapping of only one E1 channel into/from an SDH frame;
- iii. The last scenario, *global verification*, evaluates the effects of mapping/demapping multiple E1 channels into/from an SDH frame.

In the *loop verification*, depicted in Figure 7, the goal is to verify the VC12 FIFO operations (*AddDrop* circuit) and the SDH bypass path (Telecom Bus circuit). The testbench code in this scenario has also a set of VHDL assert conditions to detect exceptions and critical operations (e.g. $DELTA$ errors achieved with FIFO overflow or underflow).

Subsequently, the *BEMS verification* is performed, as is depicted in Figure 8. From the E1 input channel to the SDH frame, generated data is packed into SDH as detailed in Figure 1. The pattern analyzer program extracts E1 information from the SDH output file and compares it to the input file. At the same time, the opposite flow (from the SDH frame to the E1 output channel) is evaluated: the generated data is unpacked from SDH frame to E1. The pattern analyzer then compares the unpacked data against the SDH input file. The results allow capturing data losses and timing.

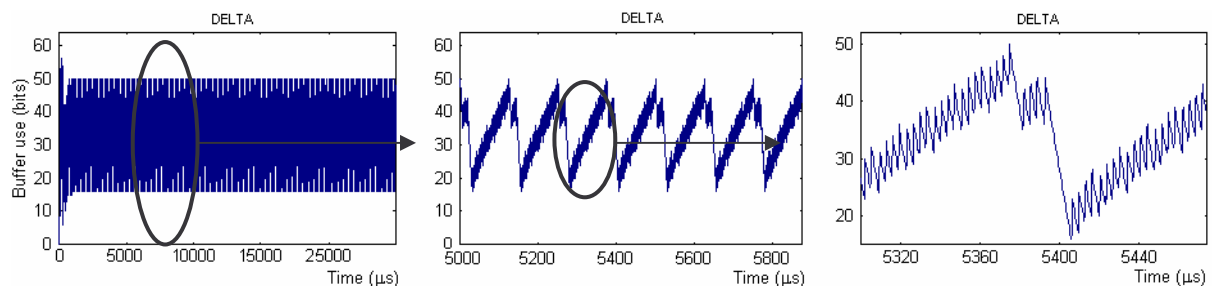


Figure 5 – $DELTA$ signal behavior for $VC12Drop$ nominal clock, for different time scales.

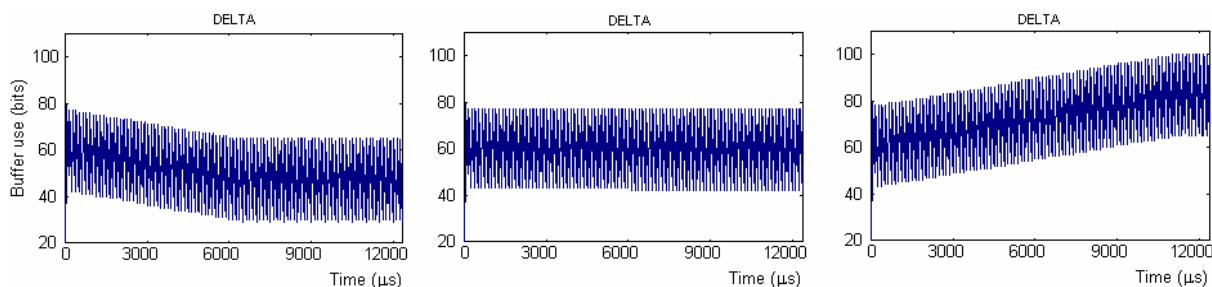


Figure 6 – $DELTA$ signal behavior for $VC12Add$ module, considering slow, nominal, and fast E1 input clocks, respectively.

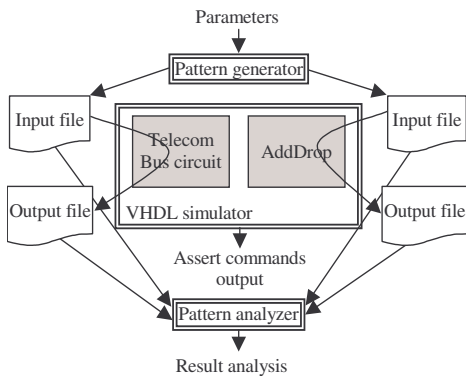


Figure 7 – Loop verification for functional validation process.

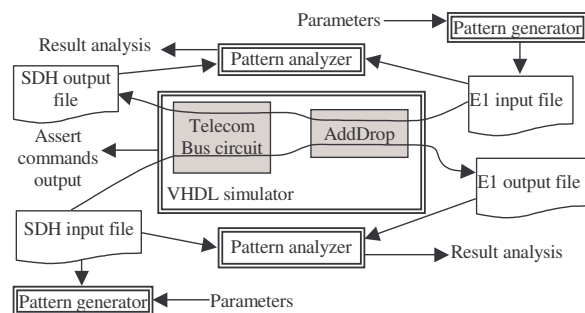


Figure 8 – BEMS functional validation process.

The *global verification* is a generalization of the *BEMS verification*, as depicted in Figure 9. The global verification evaluates the FIFO behavior due to the differences of the SDH production/consumption according to all E1 channels consumption/production. Underflow and overflow FIFO conditions are also evaluated. In this case, pattern analyzers are able to evaluate each channel separately and the join effect of all channels.

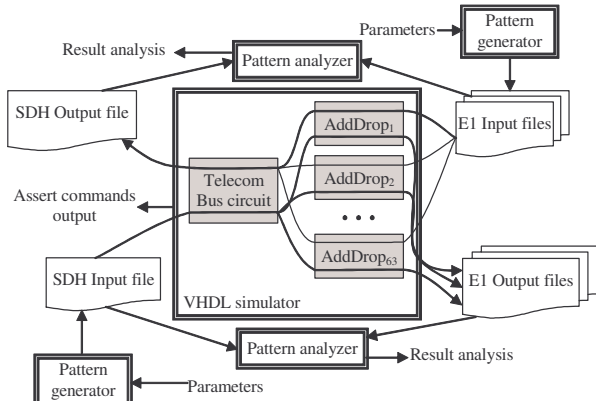


Figure 9 – EMS functional validation process.

6. Prototyping

BEMS takes 2150 lines of RTL VHDL. The code is portable, except for the FIFOs circuits, implemented using Xilinx Block SelectRAM primitives. After validation at the functional level, EMS was prototyped and validated in the VCC VW-300 prototyping platform.

This board contains a 300,000-gate Virtex FPGA. The BEMS design occupies 314 out of 3,072 slices, i.e. 10% of the FPGA device. The system fulfills all the original specification constraints. Since the SDH is relatively small and has small latency (9 clock cycles), it allows cascading several instances of it in a single system.

7. Conclusions

The main contributions of the presented work are: (i) The development of the EMS soft-core, which performs mapping and demapping of E1 channels into/from SDH; (ii) The development of a buffer technique for correct frequency justification; and (iii) the validation technique that reduces design time. Validating this otherwise small circuit proved to be a demanding task, which required the development of specific software tools. These tools allowed exploring the correctness of the generated outputs with a good degree of accuracy and coverage. This has been confirmed by running the circuit in a real world environment.

Our approach takes 9 cycles for frame propagation latency for any number of VC-12 implemented in STM-1, since all VC-12 circuits operate concurrently. Besides the low propagation latency, EMS presents small size, enabling the use of low-cost programmable devices.

The main difficulties faced during this work were: (i) The understanding of the ITU-T rules for SDH systems; and (ii) The amount of data necessary to generate and to analyze for guaranteeing a minimum degree of covering during the validation of the core.

8. References

- [1] M.-C. Chow. *Understanding SONET/SDH. Standards and Applications*. Andan Publisher, New Jersey, 1996.
- [2] F. Moraes et al. *Design and Prototyping of an E1 Drop_Insert soft core*. IEE Proc. Communications, p. 239-243, Aug. 2003.
- [3] ITU-T, *Characteristics of SDH equipment functional blocks. Recommendation G. 783*. Telecommunication Standardization Sector of ITU, 1997.
- [4] Intel Corporation. *IXF6151 28 T1/E1 Mapper*, Data Sheet, Jan. 2001.
- [5] Transwitch Inc. *Ethernet into STS-3/STM-1 SONET/SDH Mapper (TXC-04226B)*, Data Sheet, 5th ed, Jan. 2004.
- [6] X. Yongming et al. *Asynchronous Mapping of 2.048 Mbit/s Tributary into SDH VC-12*. In: ICCT, p. 817-820, May 1996.
- [7] S. Fuqiang et al. *Design of the Elastic Buffer Size for SDH Equipments*. In: ICCT'96, p. 800-804, May 1996.
- [8] R. Clauberg et al. *A Scalable Modular Architecture for SDH/SONET Technology*. In: CCCN'99, p. 442-446, Oct. 1999.
- [9] R. Peng et al. *Design and Applications of SDH Cross-Connection ASIC*. In: WCC-ICCT 2000, p. 1041-1045, Aug. 2000.
- [10] R. Silveira and W. A. M. van Noije. *Modeling an E1/TU12 Mapper for SDH Systems*. In: Proc. of 13th Symposium on Integrated Circuits and Systems Design, p. 171-176, Sept. 2000.